



**AIAA 2002-5409**

**Bi-Level Integrated System Synthesis  
(BLISS) for Concurrent and Distributed  
Processing**

Jaroslav Sobieszczanski-Sobieski  
NASA Langley Research Center  
Hampton, VA 23681-2199

Troy D. Altus, Matthew Phillips, and Robert Sandusky  
George Washington University

**9th AIAA/NASA/USAF/ISSMO  
Symposium on Multidisciplinary Analysis  
and Optimization  
September 4-6, 2002  
Atlanta, GA**

For permission to copy or republish, contact the copyright owner named on the first page.  
For AIAA-held copyright, write to AIAA Permissions Department,  
1801 Alexander Bell Drive, Suite 500, Reston, VA, 20191-4344.



# BI-LEVEL INTEGRATED SYSTEM SYNTHESIS (BLISS) FOR CONCURRENT AND DISTRIBUTED PROCESSING

Jaroslaw Sobieszczanski-Sobieski<sup>†</sup>, Troy D. Altus<sup>‡</sup>, Matthew Phillips<sup>‡</sup>, and Robert Sandusky<sup>‡</sup>

## Abstract

The paper introduces a new version of the Bi-Level Integrated System Synthesis (BLISS) methods intended for optimization of engineering systems conducted by distributed specialty groups working concurrently and using a multiprocessor computing environment. The method decomposes the overall optimization task into subtasks associated with disciplines or subsystems where the local design variables are numerous and a single, system-level optimization whose design variables are relatively few. The subtasks are fully autonomous as to their inner operations and decision making. Their purpose is to eliminate the local design variables and generate a wide spectrum of feasible designs whose behavior is represented by Response Surfaces to be accessed by a system-level optimization. It is shown that, if the problem is convex, the solution of the decomposed problem is the same as that obtained without decomposition. A simplified example of an aircraft design shows the method working as intended. The paper includes a discussion of the method merits and demerits and recommendations for further research.

## Nomenclature

AR	aspect ratio
BB	“black box”, another term for a module
c	coupling equality constraints
g	behavior constraints local to a BB
h	equality constraints tantamount to solution of analysis
ESF	engine scale factor
h	cruise altitude
L/D	lift to drag ratio
Q	system level design variables, {Xsh   Y*   w}, a subset of Z
R	flight range
RS	Response Surface, SRS – sheaf of RS
SCF	specific fuel consumption
Sref	wing reference area.
t/c	airfoil depth
ts	the cross-sectional dimensions

T	throttle setting
U	local design variables, {Xloc   Y^}, a subset of Z
U & L	upper and lower bounds
Lht	horizontal tail location coordinate
M	Mach number
w	weighting factor in suboptimization objective function
W	weight
Wt	total weight, also denoted by TOGW for take-off gross weight
Xloc	design variables local to a BB
Xsh	shared design variables affecting directly two or more Black Boxes (modules), BB
Y*	behavior variables input into a BB from other BB's
Y^	behavior variables output from a BB, some elements of Y^ are designated to be Y*
Y^_	a particular data item selected in a particular BB output to be the system objective
Z = {Xsh   Xloc   Y*   Y^}	as a vector of variables <sup>1</sup> in a not decomposed, combined analysis and optimization problem
λ	taper ratio
Λ	sweep angle
Θ	effective wing area change due to twist

### Superscripts:

a	approximate
^	output from a BB
*	input to a BB generated by the system optimizer

### Subscripts:

0	optimal
E	engine
F	fuel
HT	horizontal tail
s	system objective
T	Take-off-Gross-Weight, TOGW
w	wing

<sup>1</sup>Examples in aircraft design: Xsh – wing aspect ratio and sweep angle; Xloc – wing cover panel thickness and a composite ply orientation angle; Y\* and Y^ – an elastic deformation that alters the wing aerodynamic shape, and aerodynamic loads that cause the wing deformation.

<sup>†</sup>NASA Langley Research Center, MS 240, Hampton, VA 23681.

<sup>‡</sup>The George Washington University, the LaRC campus.

## 0. Introduction

The relentless drive of computer technology toward ever higher computing speed (Floating Point Operations Per Second, FLOPS) has enabled solution of large computational problems encountered in design of complex engineering systems, e.g., an aircraft, in a small fraction of time that was required just a few years ago. And yet, it is still not practical to solve such problems in one monolithic calculation because the number of design variables is large, the analysis is high fidelity and it consists of a system of coupled codes. Such monolithic calculation, perhaps, may never become a routine practice regardless of the computing speed available because design of complex engineering systems requires that all sorts of information ranging from experimental to computational be synthesized by human judgment. Consequently, it is a common practice to conduct a design process by collaboration of autonomous groups of specialists that retain control over their domains of expertise, and work concurrently to compress the project elapsed time.

Motivated by this, several methods have emerged to enable decomposition of the system optimization into a set of smaller tasks aligned with disciplines or physical subsystems. Recently, it became evident that the above methods are natural for computing with large numbers of concurrently operating processors (Massively Concurrent and Distributed Processing, MCDP), a new technology now developing a new computational infrastructure having a potential to support engineering design at an unprecedented level of effectiveness.

Optimization by decomposition, initiated by Ref. 1, became a key component of the Multidisciplinary Design Optimization<sup>2,3,4</sup>, and resulted in a body of literature much too large to be cited here in full. A few representative references are Refs. 5-15, and 16,17 (the last two are surveys). Their common thread consists of suboptimizations at a lower level, all coordinated by solution of a higher, system-level optimization. Their diversity lies in the means that link the two levels, in the use of various approximations, and in the definition of the objective function(s) at both levels. As to the linking, the typical approaches are the sensitivity of optimum to parameters, the system behavior sensitivity, control over the contributions a suboptimization makes to the satisfaction of constraints in another suboptimization, and imposition of targets on the suboptimization results.

The approximations commonly used in constructing surrogate linear or nonlinear models are often based on the disciplinary and system behavior derivatives.

Recently, domain-spanning approximations (DSA) such as the Response Surfaces (RS) or Neural Nets (NN) constructed with the aid of Design of Experiments (DOE) techniques became popular because they lead to repetitive but independent calculations that can be performed simultaneously to exploit the MCDP technology (e.g., Refs. 8, 11, and 18).

The choice of the objective function(s) is fundamentally important. It is also difficult because in a suboptimization the objective ought to reflect local information combined with the influence of that information on the system performance, and vice versa at the system level, while preserving the suboptimization autonomy.

A structural design of the wing is an example that illustrates considerations that underlie the choice of the objective in a subsystem optimization. Its output includes structural weight and elastic deformation. Conventionally, the weight would be the objective. However, the ultimate objective is a measure of the aircraft system performance. That performance, in general, benefits from reduction of weight and drag. Because increased structural stiffness lowers the elastic wing drag, it follows that, for the system benefit, the wing structure ought to be designed for an optimal combination of low weight and high stiffness. Furthermore, the wing structural weight may be lightened by tailoring the wing deformation to redistribution of the aerodynamic load so as to reduce the root bending moment. The resulting load distribution is, in general, different from the one that minimizes the drag.

The original version of BLISS<sup>10,19,20</sup> addressed the above considerations by computing the system sensitivity derivatives of the system objective function with respect to the subsystem design variables and, then, define a subsystem objective function as a sum of the subsystem design variables weighted by these derivatives. The subsequent versions included the use of RS<sup>18</sup> and a variant specialized for structural optimization<sup>21</sup>.

The key new concept in BLISS 2000 reported herein is a new formulation for the objective function in the subsystem optimization and the use of that optimization to control the subsystem output for the benefit of the system performance. In the new formulation, the subsystem optimization objective is a sum of the subsystem outputs, each output weighted by a coefficient that is treated as a design variable in the system-level optimization. This formulation eliminates the costly system sensitivity and optimum sensitivity analyses. It also enables representation of the subsystem optima at the system-level by the Response Surfaces prepared

“off-line” by autonomous, concurrent, and potentially distributed operations. It is expected that the specialty groups in charge of these operations work in a MDCP environment, being free to draw on any sources of information and to apply any tools of their own choice.

The remainder of the paper defines the BLISS 2000 algorithm first. It follows with a discussion and interpretation of the algorithm in qualitative terms, and offers a theoretical justification of the approach by demonstrating that the solution obtained by decomposition also solves the same problem formulated without decomposition. Remarks about the computational cost conclude the method description.

A simplified, conceptual-level design problem of a supersonic business jet illustrates the method and contributes an empirical basis to a concluding assessment of the method limitations, advantages, and recommended future developments.

### 1. The BLISS 2000 Algorithm

The object of optimization in BLISS is a generic engineering system whose representative example is an aircraft (Fig. 1). The diagram in Fig. 2 shows a few major disciplines that form a coupled system in the aircraft mathematical model. The model is comprised of modules, also called Black Boxes (BB), coupled by data exchanges.

#### 1.1. The Original Problem

For any optimization-by-decomposition scheme to be valid, the solution it generates should also solve the problem formulated without decomposition. Therefore, introduction of the BLISS algorithm begins with an all-in-one formulation that combines analysis and optimization of a generic system of which Fig. 2 is an example. It reads

FIND:	Z	1)
MINIMIZE:	$F(Z) = Y^{\wedge}(Z)$	a)
SATISFY:	$g(Z) \leq 0$ , for each BB;	b)
	$h(Z) = 0$ , for each BB;	c)
	$c(Z) = Y^* - Y^{\wedge} = 0$	d)
	$ZL \leq Z \leq ZU$ ; side constraints.	e)
OUTPUT:	$Z_0, F_0$	f)

In the above, the inequalities  $g$  represent the behavior constraints local to a BB<sup>2</sup>, and the equalities  $h$  correspond to the solution of the BB governing equations (the BB inner analysis)<sup>3</sup>. The output-input equalities  $c$

<sup>2</sup>Example: structural strength; limit on the aerodynamic pressure gradient along the wing airfoil chord.

<sup>3</sup>Example: equations of the Finite Element Analysis.

describe the system inner couplings<sup>4</sup>, and in Eq. 1 they are present for two reasons:

- The input  $Y^*$  to a particular BB is supplied by the optimizer instead of being received directly as  $Y^{\wedge}$  from another BB, where it would originate in a conventional analysis of a coupled system. Therefore, the equalities  $c$  are needed to ensure that upon convergence the optimizer-generated  $Y^*$  is the same as the corresponding  $Y^{\wedge}$ .

- The system analysis needs to be solved. Equating the corresponding pairs  $Y^*$  and  $Y^{\wedge}$  is tantamount to obtaining a solution.

The formalism of Eq. 1 is known as the Simultaneous Analysis and Design (SAND). It brings the  $h$  and  $c$  equalities under a single optimization algorithm in contrast to the more common optimization techniques that eliminate  $Y^{\wedge}$  and  $Y^*$  from  $Z$  by solving the  $h$  and  $c$  equalities for each set of the trial values of  $X_{sh}$  and  $X_{loc}$  sent out by the optimizer.

#### 1.2. BLISS 2000 Optimization by Decomposition

The BLISS 2000 algorithm separates the BB suboptimizations from the system optimization. To introduce the algorithm, this section provides a formal statement for the optimization at the BB level, followed by definition of the Response Surfaces that link the above optimization to the system-level optimization, and concludes with a formal statement for the latter. The section that follows elaborates on a rationale for the two-level procedure, its theoretical underpinnings, and the computational cost considerations.

1.2.1. BLISS Subsystem (Black Box) Optimization Problem Repeatedly Executed for a Number of Dispersed Design Points. The design variables  $Z$  are divided into the system level variables  $\{X_{sh} | Y^*\}$  and the subsystem level variables  $\{X_{loc} | Y^{\wedge}\}$  and the following optimization problem is solved for each BB:

GIVEN:	$Q = \{X_{sh}   Y^*   w\}$	2)
FIND:	$U = \{X_{loc}   Y^{\wedge}\}$	a)
MINIMIZE:	$f(U) = \sum w_i Y^{\wedge}_i$	b)
SATISFY:	$g(U) \leq 0$ , for each BB;	c)
	$h(U) = 0$ , for each BB;	d)
	$UL \leq U \leq UU$	e)
OUTPUT:	$Y^{\wedge}_0$ , and $X_{loc0}$	f)

The weighting coefficients  $w_i$  in  $f$ , one coefficient per element of the output vector  $Y^{\wedge}$ , are appended to the system-level variables in order to link the BB

<sup>4</sup>Example: the aerodynamic loads that deform a wing should be computed for the wing deformed due to those loads.

suboptimization to the system optimization. For reasons to be explained later, the constrained minimum of  $f$  is not passed to the system optimization; only the  $Y^a_0$  values are transferred.

The Eq. 2 optimization is executed at a number of points dispersed in the Q-space that belongs to the BB. The formal Design of Experiments (DOE) techniques may aid in forming the dispersal pattern to achieve a reasonable coverage of the domain. From the system perspective the solution method choice is immaterial and that choice need not be the same for all the BB's. The specialists in charge of a BB are free to use any suitable method, even experiments, or a guess.

### 1.3. Approximate Model for Optimized Subsystems

Next, a Response Surface (RS) is fitted (or a Neural Net may be trained) to each of the elements of  $Y^a_0$ . Regarding each RS as a leaf in a sheaf, the resulting data base is a Sheaf of RS (SRS). It constitutes an approximate model in the Q space of the BB optimized using the U coordinates, so that

$$Y^a_0 = Y^a_0(\text{SRS}(Q)); \text{ and } U^a_0 = U^a_0(\text{SRS}(Q)) \quad \text{for } QL \leq Q \leq QU \quad (3)$$

In the above, the superscript "a" denotes approximate values, and the notation such as  $Y^a_0 = Y^a_0(\text{SRS}(Q))$  means that the approximate values are retrieved from precomputed SRS. The bounds on Q, QL and QU, are the best estimates, accounting for any side constraints and, also, incorporating the move limits necessary for the ensuing inter-level iteration. The SRS may be called a domain approximation because it covers the entire Q space within the above bounds.

In principle, a similar RS approximation could be constructed for each element of  $Xloc_0$ . However, the resulting volume of data to be stored may be so large as to make this impractical. If so,  $Xloc_0$  can be regenerated in a manner discussed in Sec. 1.5.

The computational cost of an RS is proportional to the number of points to be evaluated which grows with the number of variables. It is desirable, therefore, to reduce the number of variables in Q by a condensation technique. Such condensation is imperative for those variables in Q that represent field quantities, e.g., the field of the pressure loads distributed over the wing, and the corresponding field of displacements. A condensation of a field data may be accomplished by defining the field variable P as

$$P = P(p) \quad (4)$$

where the number of the parameters p is made as small as possible<sup>5</sup>. The p parameters enter the procedures as if they were  $Y^*$  and  $Y^a$ , each p being represented by its own RS.

When all the BBs are optimized, the functions in Eq. 3 for each BB are available to the system optimization that executes next.

### 1.4. BLISS System Optimization

The system-level optimization problem is:

GIVEN:	a set of SRS, one for each BB	5)
FIND:	{Q} = {Xsh   Y*   w}	a)
MINIMIZE:	F(Q) = $Y^a_{ik}$	b)
SATISFY:	c = (Y*, Y^a(Q)) = 0;	c)
	QL <= Q <= QU	d)
OUTPUT:	Q <sub>0</sub> , F <sub>0</sub>	e)

Eq. 5 may be solved by any optimization technique, the data for  $F(Q) = Y^a_{ik}$  and  $c(Q)$  for any Q being retrieved, as per Eq. 3, from the SRS data base. The efficiency of the technique is not critical because that data retrieval is nearly instantaneous. As in Eq. 1,  $c = 0$  in the above is tantamount to a system analysis, thus the BLISS algorithm implements, in effect, the Simultaneous Analysis and Design method at the system level, but not necessarily at the BB-level where the choice of methods for solving Eq. 2 is autonomous.

### 1.5. BLISS Iterative Procedure

The retrieval of data from SRS (Eq. 3) in a non-linear system, is burdened with an error,  $\epsilon = Y^a_{ik} - Y^a_{ik}$  whose control requires iteration between Eq. 2 and 5 involving move limits incorporated in QL and QU in Eq. 5d. These move limits are being adjusted in each iteration and, occasionally, they may push against the SRS bounds. That may require the addition of new points at which to solve Eq. 2, and refitting of the SRS.

The iterative procedure may also include at its conclusion a retrieval of the optimal Xloc whose elements might have been stored in form of SRS. If the volume of such storage is prohibitively large, one may choose to regenerate the Xloc instead by repeating Eq. 2 while substituting the latest optimal {Xsh|Y\*|Y^a|w}. However, such regeneration is tantamount to a partial execution of one additional iteration of the entire BLISS procedure, therefore, some of the consistency between Eq. 2 and 5 is lost resulting in an additional error.

<sup>5</sup> Example: Displacements normal to a wing may be represented by a function in the chordwise and spanwise coordinates. Making that function linear chordwise and cubic spanwise reduces the number of parameters p to only 5.

The overall, step-by-step recipe for BLISS 2000 optimization algorithm is now written as:

0. Start
1. Initialize Xsh and w, and the U and L bounds
2. Solve equations
  - a.  $h = 0$
  - b.  $c = 0$
  - i.e., perform an initial System Analysis to obtain starting values of  $Y^*$  to be considered in establishing the RS bounds in the next step.
3. Approximate model development for each BB. This step may be done simultaneously for all the BB's.
  - a. Reduce dimensionality of the Q space per Eq. 4.
  - b. Disperse by a suitable DOE technique a minimum number of points required to define an RS approximation in the BB Q-space, bounded by QL and QU.
  - c. Solve Eq. 2 in subspace U at the above points in space Q. This may be done simultaneously for all the points within Q.
  - d. Fit a Sheaf of Response Surfaces (SRS) to the results of #3b.
  - e. Verify quality SRS by random sampling. If needed, add new points and discard old points, and use least square fit (or an equivalent technique) to improve the SRS quality.
  - f. After each system optimization (#4 below): shift, extend or shrink the intervals (QL,QU) to avoid excursions beyond the SRS bounds and to maintain the approximation quality.
4. Solve system optimization Eq. 5 in space Q, accessing the SRS data per Eq. 3.
5. Check the termination criteria: Exit, or Repeat from #3 using SRS already available or updated per #3e and f.
6. Retrieve the optimal Xloc
7. Stop

One execution of the procedure from #3 through 5 is called the major iteration or the cycle.

## 2. The BLISS Rationale, Theoretical Justification, and Computational Effort

Having the BLISS 2000 algorithm defined, attention now turns to its underlying rationale, theoretical justification, and the computational cost of the method.

### 2.1. Qualitative Discussion of Rationale

The system objective  $F$  is computed as an element of  $Y^\wedge$  in one of the BB's. The  $Y^\wedge$  from any BB influences  $F$  because of the couplings  $c$ . However, one does not initially know the strength or even the sign of these influences. *The suboptimization task for a BB is, then, to develop a wide choice of the BB designs, each*

*having a different set of outputs  $Y^\wedge$  and each being feasible with respect to the local constraints  $h$  and  $g$ .*

That task is accomplished by using the BB optimization in Eq. 2 merely as a tool to control the BB output  $Y^\wedge$ , the weighting coefficients  $w$  in Eq. 2b acting as parameters of that control. Replacement of the set of the original vector of the system-level design variables  $\{Xsh | Y^*\}$  with the vector  $\{Xsh | Y^* | w\}$ , that includes  $w$  as additional design variables, results in a range of the BB designs,  $\{Xloc | Y^\wedge\}$ , at each point in the  $\{Xsh | Y^*\}$  space, all the BB designs being feasible. Availability of a wide choice of these designs in form of SRS enables the system-level optimization in Eq. 5 to find a set of the BB designs that are compatible with respect to  $c = 0$ , and generate outputs  $Y^\wedge$  whose collective influence minimizes the objective Eq. 5b.

The above approach to BB suboptimization rests on the concept of optimizing a component to attain a desired response. That concept may be illustrated by a simple example of a cantilever beam of length  $L$  and a square cross-section of the side length "a", loaded by a force  $P$  at the tip. Consider the beam volume  $V = a^2 L$ , the tip displacement  $d = PL^3/(3EI)$ , where  $I = a^4/12$ , and the bending stress at the root  $s = 6PL/a^3$ , as three outputs that define the beam response. Let's now choose arbitrarily two items out of these three outputs,  $V$  and  $d$  to show how their control may be achieved using a weighting coefficient  $w$  inserted in the composite objective function in the optimization formulated as

GIVEN: constants of  $E$  – Young's modulus;  
 $L$  – beam length; and  $s_a$  = allowable stress;  
 FIND: dimension  $a$   
 MINIMIZE:  $f = V + w d$   
 SATISFY:  $s \leq s_a$

Substituting the formulas for  $V$  and  $d$  into  $f$ , and setting  $df/dw = 0$  yields

$$a_0 = \max(C_1 w^{(1/6)}, (6PL/s_a^{(1/3)}));$$

$$V_0 = C_2 w^{(1/3)}; d_0 = C_3 w^{(2/3)}; V_0/d_0 = w^{(-1/3)}$$

where  $C_1$  and  $C_2$  are constants comprising parameters  $P$ ,  $L$ ,  $E$ , and  $s_a$ . This solution shows how  $w$  controls the ratio of the two response outputs.

One should acknowledge that a similar control of the outputs may be accomplished by other means. For example, the Collaborative Optimization method (e.g., Refs. 9 and 11) achieves that control by imposing targets  $Y^1$  on the output variables  $Y^\wedge$  and minimizing the

discrepancy  $f = \sum (Y^l - Y^a)^2$  between  $Y^l$  and  $Y^a$  as the objective function in Eq. 2 to generate the optimal  $Y^a$ . Unlike BLISS 2000, CO includes the Xsh inputs as design variables in U in Eq. 2 together with Xloc, and imposes the targets on Xsh to generate optimal Xsh in addition to optimal Xloc. The w variables are absent in the CO formulation as their role is assigned, in effect, to the target variables. In both the CO version reported in Ref. 11 and in BLISS 2000 herein, the optimal results are stored for the use in the system-level optimization in form of RS.

The formulation with targets  $Y^l$  in place of w appears to be a viable alternative in BLISS. It could be combined with stating the objective function in Eq. 2b in a Goal Programming style<sup>22</sup> as  $f = \sum (d^+ + d^-)$  where  $d^+$  and  $d^-$  measure the overachievement or underachievement of the targets. Examination of merits of the above alternatives is beyond the scope of this paper.

## 2.2. Theoretical Justification

The following chain reasoning shows that a solution of the set of Eq. 2 and 5 satisfies the original problem described by Eq. 1:

1. Solution of Eq. 2 satisfies constraints  $h = 0$  and  $g \leq 0$  present in Eq. 1.
2. Solution of Eq. 5 satisfies constraints  $c = 0$  present in Eq. 1.
3. Solution of Eq. 2 and the subsequent generation of SRS, Eq. 3, renders  $Y^a = Y^a_0(\{Xsh | Y^* | w\})$ , and  $Xloc = Xloc(\{Xsh | Y^* | w\})$ . If the problem is convex, the above relations are unique for each vector element (single-valued functions).
4. Owing to the above relations, the space of  $(Xsh | Y^* | w)$ , searched in Eq. 5, maps uniquely on the original space of  $\{Xsh | Xloc | Y^* | Y^a\}$ , searched in Eq. 1, because Xsh and  $Y^*$  define both spaces, and Xloc and  $Y^a$  are turned into functions of  $\{Xsh | Y^* | w\}$  per #3 above.
5. The existence of the mapping of the space of Eq. 5 onto the space of Eq. 1 together with satisfaction of the constraints g, h, and c, assures that the solution of the sequence Eqs. 2, 3, 5 satisfies also Eq. 1, subject to the caveats that follow.

Caveat 1: If the problem of Eq. 1 has local minima (a non-convex problem), then the solutions of Eqs. 2, 3, 5 may not arrive at the solution of Eq. 1 even when starting from the same initial point because the algorithmic differences may result in different search path being traced.

Caveat 2: If the SRS were error free and wide enough to contain the solution to Eq. 5, a single execution of the sequence Eq. 2 - Eq. 5 would suffice.

Otherwise, the sequence has to include the SRS updates and be iterated to reduce the error  $\varepsilon = Y^a_0 - Y^a$ .

## 2.3. Computational Effort

In BLISS 2000 the SRS data may be generated simultaneously and independently at distributed sites. Furthermore, this data generation does not require any modifications to the existing codes. Each code involved in Eq. 2 may simply be replicated over many processors and executed with different inputs Q. It is a classic case of the so-called coarse-grain parallelism in computing. Thus, there is no additional up-front reprogramming cost although there is some cost penalty in communication and bookkeeping.

The BLISS computational cost as measured by the elapsed time is primarily in repetitive execution of Eq. 2 that for a large-scale problem overshadows the cost of Eq. 5. For a quadratic polynomial RS in the Q space of N dimensions the minimum number of points is  $M \approx O(N^2/2)$ . However, one may choose to increase the number of points to  $M_2 = M_1 M$ ,  $M_1 > 1$ , to improve the RS accuracy, in which case the RS may be fitted by means of the method of least squares (or an equivalent) with the added benefit of smoothing that lessens the probability of entrapment in a local minimum. Furthermore, the procedure of solving the sequence Eq. 2-5 requires a problem-dependent number of cycles, NQ, so the total number of points increases to  $O(NQ, N^2/2)$ . Consequently, the quadratic RS may be impractical beyond  $N = 12$  to 20 in applications where the elapsed computing time per point is substantial and all calculations are sequential. Therefore, the use of BLISS 2000 in large-scale applications depends on the technology of Massively Concurrent and Distributed Processing (MCDP) for compressing M operations into the elapsed time of one. Such compression not only extends radically the RS limits of practicality now but also provides for their further relaxation with the progress of the MCDP technology.

Another way to circumvent the "curse of dimensionality" is to settle for a linear form of RS that makes  $M \approx O(N)$ . Then, the iterative process of solving the sequence of Eq. 2-5 becomes similar to the sequential linear programming (or its more refined version of the sequential approximate programming<sup>23</sup>). Many years of experience of the above practice with diverse applications indicates that the number of times, NA, the approximations must be refreshed in these sequential processes depends on the class of application but stays independent of N and reasonably small within that class. For instance<sup>23,24</sup>, in structural optimization dominated by membrane stresses, typically,  $NA = 5$

to 10, and it rises to 20 to 30 where bending is prevalent<sup>24</sup>. The total number of points to be evaluated in such a process is of  $O(NA, N)$ . A recent numerical evidence attesting effectiveness of the linear RS in a sequential approximate optimization is provided in Refs. 25 and 26 where the linear RS is used in an MCDP environment, and in Ref. 11 where it is an enhancement in CO.

Consequently, substituting NA and NQ expected for a particular BB and checking whether  $O(NA, N) < O(NQ, N^2/2)$  is true constitute an important factor in the choice between the quadratic or linear RS for that particular BB.

Another factor is an organizational one. The RS refreshing operations (Eq. 2) need to be completed at the outset of each cycle so a degree of coordination among the groups performing these operations is necessary. The cost of that coordination adds to the cost of the entire procedure. On that score, the higher order RS are superior because NQ is likely to be smaller than NA, hence, the coordination is needed less often.

### 3. A Numerical Example

A Supersonic Business Jet (SBJ) shown in Fig. 1 is a numerical example for the BLISS 2000 algorithm. It is the same case that was used in Refs. 10, 19, and 20 in the overall layout, but sufficiently different in detail to make the results incomparable. Ref. 27 completely documents the present case and its relation to the previous one. The model of the aircraft system is reduced to four BB's shown in Fig. 3 which shows also their data exchanges in a standard format of the data dependence matrix, also known as the n-square format. That format depicts the modules strung along the diagonal. Each module accepts input vertically from above or below, and outputs horizontally, left or right.

The BB-structures is a plate representation of the wingbox, connected to a rigid beam model of the fuselage to which a horizontal tail is attached. The wingbox strength (stress and local buckling constraints) and stiffness accounts for the wing sandwich covers. The wingbox volume is the fuel tank volume. The BB-aerodynamics calculates lift distribution sensitive to the wing trapezoidal geometry changes and to the local variations of the angle of attack generated by an aerodynamic twist and by the elastic deformation. This BB includes also computation of the drag accounting for the wave drag, and accounts for the trim constraints by performing a trim analysis involving the tail volume. The BB-power simulates the propulsion by interpolation of a look-up table that contains data on the thrust

and specific fuel consumption as functions of the Mach number and altitude. Finally, the BB-performance computes the flight range by the Breguet equation.

The system objective is the range under constant TOGW. The system design variables govern the geometry and include the Mach number and altitude. The local variables are: in structures, the cross-sectional dimensions of the sandwich wing covers at several locations over the wing; in aerodynamics, the horizontal tail sweep angle, wing and horizontal tail location coordinates; and in Power the throttle setting. Table 1 provides more information for the input, output, and the RS statistics for the BB's. The field variables of aerodynamic loads and the wing twist were condensed to their distribution parameters.

The n-square diagram in Fig. 4 corresponds to the one in Fig. 3, except that it shows the system level optimizer as the sole input source for all BB's. Each BB in that diagram is now a SRS representation of that BB suboptimization. The c-constraints occur at the data channel intersections marked by circles.

The design points for construction of RS are placed using the D-optimal technique from the Design of Experiments methodology. The RS are quadratic polynomials. As the procedure progresses, they are being periodically updated in order to keep them centered in the Q space around the latest optimal solution, to prevent the optimal solution to remain lodged against the RS boundary, and to reduce the RS span as the procedure homes on the system optimum.

The BB analysis codes are in FORTRAN, while MATLAB Optimization Toolbox is the tool for optimization at the BB level and the system level. The main program that executes the entire procedure is also written in MATLAB.

In this example, BLISS 2000 manipulates the system and local design variables so as to adjust the wing structural weight, engine weight, aerodynamic drag, fuel volume, and fuel consumption so as to maximize the range.

The results are very voluminous<sup>27</sup> so only a small but representative sample is displayed herein in Table 2 that shows how the objective and a few local and system variables were changing over the BLISS cycles. In this case, the result variations diminish to the level commensurate with the accuracy of the analysis typically after 6 or 8 cycles, hence the number of cycles is set to ten. The data are shown normalized by their values obtained in Iteration 10; the latter are also presented in the right-most column. The BB internal optimization histograms are not shown because the local

optimization performance has no bearing on the overall convergence of the method owing to the BB autonomy.

Table 2 includes the objective function of the flight range. At a cycle number, the table shows the range predicted by RS,  $R_{rs}$ , used in that cycle and the range actually computed,  $R_{an}$ , at the outset of the next cycle. The two values converge into a complete agreement in five cycles. The convergence is oscillatory partially due to the initial differences of the RS-predicted and actual values and partially due to the search of a compromise among the multitude of trade-offs at two levels accounting for constraints.

The method transforms the initially infeasible design to one where all the constraints are satisfied and the objective is improved by 47 %. That improvement value is significant merely as a metric of performance of the method in this particular test but the range final value must not be interpreted as realistic. The analysis scope and fidelity, and completeness of the set of variables and constraints would have to be much higher for such interpretation.

#### 4. Concluding Remarks

BLISS 2000 method is intended specifically for support of design projects that require collaboration of autonomous, distributed specialty groups and exploitation of the Massively Concurrent and Distributed Processing (MCDP) technology.

The method decomposes the overall system optimization into suboptimizations associated with the specialty groups. Response Surfaces approximate the suboptimization optimal results in the coordinating, system-level optimization that, provably, leads to the same solution, subject to the approximation errors, that would have been obtained without decomposition. An example of a simplified, conceptual design of an aircraft illustrates the method.

The proof-of-the-concept testing reported herein showed BLISS performing as intended. Preliminary assessment of the method merits and demerits follows.

The BLISS 2000 performance depends on the quality of approximation of the subsystem optimization results as functions of the system-level design variables and coupling variables. The number of times these approximation need to be updated and the number of new design points in these updates are problem dependent. Assessment of the approximation errors is a factor in choosing between the linear or higher order approximations for the individual modules, and in deciding on the frequency of the approximation updates.

BLISS 2000 tends to generate a large amount of computing and voluminous intermediate data. On the other hand it simplifies the entire procedure conceptually to reduce the human effort and time. This appears to be a cost-effective trade-off, considering that computing gets less expensive all the time while the labor costs grow. The elapsed time consumed by the BLISS-generated computation is compressed by the MCDP technology. That technology has established itself the future of computing in science and engineering and BLISS being intrinsically coarse-grain parallel is posed to exploit that technology and grow with it.

The BLISS method performs best in applications that decompose into subsystems with numerous local design variables and constraints leaving a relatively small number of design variables at the system level. A nearly complete autonomy of the operations in the subsystems is the key BLISS advantage that enables full utilization of the local knowledge and control over the budget and task time.

The system level-optimization benefits from nearly instantaneous response of the approximations, therefore, it does not call for any particularly high sophistication or efficiency of the search technique.

Further research is recommended to determine how much BLISS could reduce the elapsed project time, and whether it would benefit from the alternative formulations of the module-level optimization, use of the linear instead of quadratic form of the Response Surfaces, and approximations other than Response Surfaces. It is through generation of experience in large-scale practical applications that guidelines for further gains in efficiency and accuracy will develop.

#### 5. References

1. Schmit, L.A.; and Ramanathan, R.K.: Multilevel Approach to Minimum Weight Design Including Buckling Constraints. AIAA J., Vol.16, No.2, pp.97-104.
2. Sobieszczanski-Sobieski, J.: Multidisciplinary Design Optimization; an Emerging New Engineering Discipline; in Advances in Structural Optimization, Jose Herskovits (Ed.); Kluwer Academic Publ. 1995; pp.483-496.
3. Cramer, E.J.; Dennis, J.; Frank, P.D.; Lewis, R. M.; and Shubin, G.R.: Problem Formulation for Multidisciplinary Design Optimization; SIAM Journal on Optimization, Vol.4, No.4, 1994, pp.754-776.

4. Alexandrov, N. (ed); and Hussaini, M.Y.(ed): Multidisciplinary design Optimization – State of the Art; Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization, SIAM, 1997.
5. Sobieszczanski-Sobieski, J.: A Linear Decomposition Method for Large Optimization Problems; NASA TM-83248, 1982.
6. Sobieszczanski-Sobieski, J.: Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems; Proceedings of NASA/USAF Symposium on Multidisciplinary Optimization; NASA CP-3031, Sept. 1988.
7. Renaud, J., and Gabriele, G.: Improved Coordination in Nonherarchic System Optimization; AIAA J., Vol.31, No.12, 1993, pp.2267-2373.
8. Wujek, B.A.; Renaud, J.E.; Batill, S.M.; and Brockman, J.B.: Design Flow Management and Multidisciplinary Design Optimization in Application to Aircraft Concept Sizing. 34<sup>th</sup> AIAA Aerospace Sciences Meeting, Reno, Nevada, Jan. 1996, Paper No. AIAA 96-0713.
9. Braun, R.D., and Kroo, I.M.: Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment, in Multidisciplinary Design Optimization: State of the Art, Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization, ed.: N. Alexandrov and M.Y. Hussaini, 1997, SIAM; pp.98-116.
10. Sobieszczanski-Sobieski, J.; Agte, J.S.; and Sandusky Jr., R.R.: Bilevel Integrated System Synthesis; AIAA J., Vol.38, No.1, Jan. 2000, pp.164-172.
11. Sobieski, I.P.; and Kroo, I.: Collaborative Optimization Using Response Surface Estimation; AIAA J., Vol.38, No.10, Oct. 2000, pp.1931-1938.
12. Papalambros, P; Michelena, N.: Model-based Partitioning in Optimal Design of Large Engineering Systems; in Multidisciplinary Design Optimization: State of the Art, Proceedings of the ICASE/NASA Langley Workshop on Multidisciplinary Design Optimization, ed.: N. Alexandrov and M.Y. Hussaini, 1997, SIAM; pp.98-116.
13. Wagner, T.; and Papalambros, P.: A General Framework for Decomposition Analysis in Optimal Design; in Advances in Design Automation. (ed. Gilmore, B.J.) Vol.2, pp.315-325; ASME, NY., 1993.
14. Kodiyalam, S.; Sobieszczanski-Sobieski, J.: "Multidisciplinary design optimization - some formal methods, framework requirements, and application to vehicle design"; International Journal for Vehicle Design, Vol.25, No.1&2 Special Issue 2000, pp.3-22.
15. Sobieszczanski-Sobieski, J.: Optimization by Decomposition in Structural and Multidisciplinary Optimization; in Optimization of Large Structural Systems, Vol.1, pp.197-233, sec.1; (ed. Rozvany, G.I.N.); NATO ASI Series: Series E: Applied Sciences – Vol.231. Kluwer Publ. 1993.
16. Balling, R.J.; and Sobieszczanski-Sobieski, J.: Optimization of Coupled Systems: A Critical Overview of Approaches; AIAA J., Vol.34, No.1, pp.6-17; 1996.
17. Sobieszczanski-Sobieski, J., and Haftka, R.T.: Multidisciplinary Design Optimization: Survey of Recent Developments; Structural Optimization, Vol.14, No.1, Jan. 1997, pp.1-23.
18. Kodiyalam, S.; and Sobieszczanski-Sobieski, J.: Bi-Level Integrated System Synthesis with Response Surfaces; AIAA J., Vol.38, No.8, Aug. 2000, pp.1479-1488.
19. Agte, J.S.; Sobieszczanski-Sobieski, J.; and Sandusky, R.: Supersonic Business Jet Design Through Bi-Level Integrated System Synthesis; paper SAE 1999-01-5622; The SAE World Aviation Congress, San Francisco, CA., Oct. 19-21, 1999.
20. Sobieszczanski-Sobieski, J.; Emiley, M.S.; Agte, J.; and Sandusky, R., Jr.: Advancement of Bi-level Integrated System Synthesis (BLISS); AIAA 2000-0421; AIAA 38<sup>th</sup> Aerospace Sciences Meeting, Reno, Jan. 2000; also published as NASA TM 2000-210305.
21. Sobieszczanski-Sobieski, J.; Kodiyalam, S.: BLISS/S: A New Method for Two-Level Structural Optimization; AIAA 99-1345; and in Structural and Multidisciplinary Optimization j., 21/1 pp.1-11.
22. Rao, S. S.: Engineering Optimization – Theory and Practice; J. Wiley & Sons, 1996, Ch. 12.3.6.
23. Haftka, R. T.; and Gurdal, Z.: Elements of Structural Optimization; Kluwer Academic Publ., 1992, Ch. 6.4., and example p.194.
24. Schmit, L. A.: Structural Synthesis - Genesis and Development; AIAA J., Vol.19, No.10, Oct. 1981, pp.1249-1263.
25. Yang, R-J; Gu, L.; Tho, C. (Ford Motor Co, Dearborn, MI); and Sobieszczanski-Sobieski, J.: Multidisciplinary Design Optimization of a Full

Vehicle with High Performance Computing. AIAA-2001-1273; 42<sup>nd</sup> AIAA Structures, Dynamics, and Materials Conference; Seattle, WA; April 2001.

26. Sobieszcanski-Sobieski, J.; Kodiyalam, S.; and Yang, R.-J.: Optimization of a Car Body under Constraints of Noise, Vibration, and Harshness

(NVH), and Crash; AIAA 2000-1521. Journal publication : Structural Optimization; Vol.22, No.4; November 2001; pp.295-306.

27. Altus, T.D.: A Response Surface Methodology for Bi-Level Integrated System Synthesis (BLISS); NASA/CR-2002-211652; NASA Langley Research Center.

Table 1. Variables Z, Y\*, Y^ and dimensionality data for RS

BB	Output Y^	Input Variables			Number of Inputs	Number of Points
		Z	Y*	w		
1. Structures	1. $W_T$ 2. $W_F$ 3. $\Theta$	$t/c, AR_W, \Lambda_W,$ $S_{REF}, S_{HT},$ $AR_{HT}, \lambda$	$L, W_E$	$w1(2)$ $w1(3)$	11	78
2. Aero	4. L 5. D 6. L/D	$t/c, h, M, AR_W,$ $\Lambda_W, S_{REF}, S_{HT},$ $AR_{HT}, \lambda$	$W_T, \Theta,$ ESF	$w2(1)$ $w2(3)$	14	120
3. Power	1. SCF 2. $W_E$ 3. ESF	$h, M$	D	$w3(1)$ $w3(3)$	5	21
4. Performance	1. Range	$h, M$	$W_T,$ $W_F,$ L/D, SFC		6	28

Table 2. A sample of data changing over 10 BLISS cycles

Cycle	0	1	2	3	4	5	6	7	8	9	10	Value from Cycle 10
Normalized by Value from Cycle 10												
M	1.08	1.01	1.07	1.10	1.05	1.03	1.00	0.99	1.00	0.99	1.00	1.663
TOGW (lb)	1.06	1.11	1.19	1.12	1.05	1.02	1.02	1.01	1.01	1.00	1.00	31342
Sref (ft^2)	0.69	0.35	0.77	0.83	0.77	0.72	0.85	0.86	0.93	0.98	1.00	578
ESF	1.94	1.22	0.90	1.05	1.10	1.03	0.97	1.00	1.03	1.01	1.00	0.517
t/c	2.16	2.89	2.89	1.60	1.69	1.55	1.31	1.26	1.12	1.02	1.00	0.035
ts1(in) <sup>a</sup>	1.04	1.03	1.05	1.07	1.04	1.00	0.98	1.00	0.99	0.99	1.00	3.840
T (%)	1.70	1.03	1.52	1.22	1.10	1.07	1.01	0.98	1.00	1.00	1.00	20.6
Lht (%MAC)	0.71	0.61	1.00	1.00	0.60	0.76	1.00	1.00	1.00	1.00	1.00	350
w1	0.00	0.00	0.48	1.19	1.90	1.67	2.14	1.43	1.10	1.00	1.00	0.210
w3	0.00	1.79	1.61	1.02	1.16	0.89	1.25	1.02	1.00	1.02	1.00	0.560
Rrs		0.77	0.93	1.20	0.76	0.91	0.97	0.99	0.98	0.99	1.00	5247
Ran(nm)		0.68	0.91	1.07	0.77	0.91	0.97	0.99	0.98	0.99	1.00	5247

<sup>a</sup>sandwich caliper thickness of the wing upper cover inboard segment.

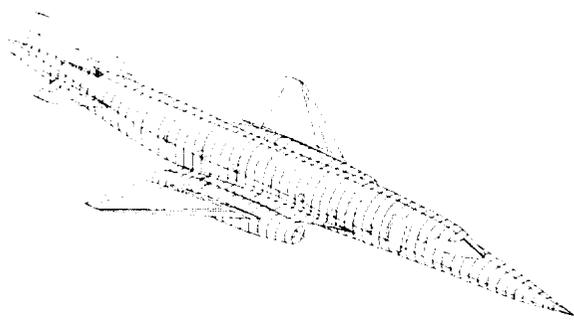


Fig. 1. A generic Supersonic Business Jet as a test case in BLISS.

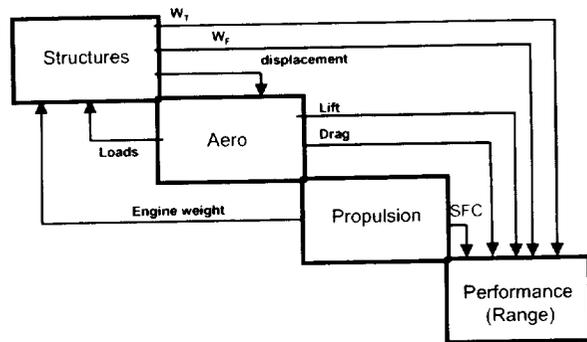


Fig. 3. Test case coupling data flow.

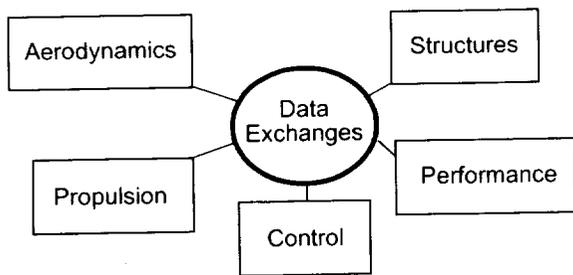


Fig. 2. An example of a few principal disciplines in an aircraft treated as a system.

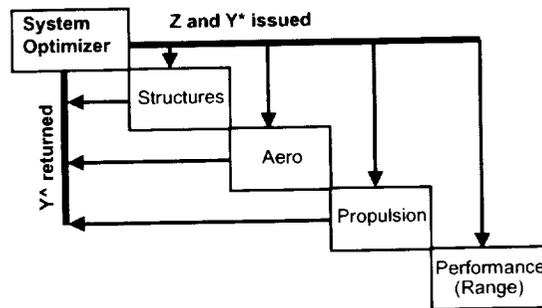


Fig. 4. System optimizer is a source of all BB inputs Z and Y\*. BB's return Y^ to the optimizer (Table 1 shows Y^, Z, and Y\*).





